1.绪论

1.1选题意义与应用背景

随着互联网的普及和Wifi技术的发展，无线/移动通信网络已经成为当今社会的基础设施，广泛地应用于政府、金融、军事等领域。统计显示，截止到2017年12月，中国网民规模达7.72亿，互联网普及率达到55.8%【2018年第41次中国互联网络发展状况统计】；与此同时，随着支付宝和微信支付等线下手机支付方式的普及，我国网民在线下消费中使用手机支付比例由2016年的50.3%提升至65.5%。无线通信在日常生活中发挥着不可替代的作用，已成为现代社会不可或缺的一部分。然而，无线网络的普及在给人们生活带来便利的同时，也存在很多安全隐患。相比于有线网络，无线网络由于其开放性更容易遭受到恶意攻击。隐私泄露、病毒入侵，网络欺骗、假冒攻击等问题时刻威胁着网络用户的安全。

传统的保护无线网络安全的方法通常是基于比特层面的(即OSI七层模型中物理层以上的层次),通过设计基于密码机制的安全协议来实现对数据完整性和机密性的保护以及提供通信双方身份的认证【无线通信设备的射频指纹提取与识别方法】。然而,密码机制存在密钥泄露的风险，无线网络安全协议也通常会存在安全漏洞【基于射频指纹的无线网络物理层认证关键技术研究】。例如,IEEE 820.11 无线局域网(WLAN)最初的有线等效加密(WEP)协议易受统计分析攻击【同上，2】,虽然此后升级为WPA和WPA2,但其口令句可以被恢复,仍然存在着各种各样的安全问题【同上，3】。此外,一旦密钥泄露，将会给用户造成难以预计的损失。然而值得庆幸的是，攻击者可以通过多种方式获取用户的账号密码，甚至模拟用户的操作，但却很难伪造出与用户相似的设备特征。因此，在无法识别无线设备用户的情况下，可以从无线设备着手，分析其独特的行为模式特征，准确识别无线设备。

在过去的十几年里，无线网络设备指纹识别问题得到了国内外广泛的关注。研究者们尝试寻找设备独一无二的属性，这些属性就如同生物技术中的基因或者指纹一样，可以唯一地标识设备，且独立于用户在设备上使用的网路协议和应用类型、不随时间变化、不易被篡改或者伪造。许多研究者提出可以从操作系统、MAC地址、浏览器等信息中找到包含设备身份信息的特征【基于软件的参考文献】，还有一些研究者认为不同型号的设备（即便是相同型号的设备）在处理器、网络适配器、DMA控制器等硬件上存在差异，而这些差异会在网络流量或通信信号等信息中反映出来，构成设备的特征指纹【基于硬件的参考文献】。

无线网络设备指纹识别技术既可以单独运作，也可以作为传统无线网络识别机制的辅助手段，提高无线网络的安全性能。选择这一研究方向的实际意义在于寻找一种被动式的无线设备指纹识别方案，通过被动地捕获网络流量，从中提取出每个设备的特征指纹，进而实现对网络设备的精确识别。这一技术可应用于入侵网络检测、接入检测、身份认证和可疑用户的监测和追踪等方面，极大地保障了无线网络的安全。

可应用于接入网络检测和入侵检测，保障无线网络安全。无线局域网中的“钓鱼AP”严重威胁着无线网络的安全。攻击者在公共场合架设一个伪装的无线接入点，设置与合法授权AP相同的服务集标识，使得受害者误连接上钓鱼AP，从而进一步展开窃取密码等攻击。因此,人们急需寻找一种能够有效识别授权用户和非授权用户的安全机制,从而降低来自恶意用户的潜在威胁。

1.2 国内外研究现状

“指纹”最早是指标识个体的生物特征识别技术【**Accelprint\_NDSS14**，Information fusion in biometrics】【P. Tuyls and J. Goseling,“Capacity and examples of templateprotectingbiometric authentication systems,”】。这一概念早在上世纪60年代就被应用于设备识别，当时研究者们开发出可观测到信号特征的系统来区分雷达【K. Talbot, P. Duley, and M. Hyatt, “Specic emitter identification andverification,”】。此后类似的技术被用于蜂窝网络中的发射器识别【M. Reizenman, “Cellular security: better, but foes still lurk,”】【L. Langley, “Specific emitter identification (sei) and classical parameter fusion technology,”】。近年来，随着对无线网络设备研究的深入，对无线/移动网络设备“指纹”的分析技术也越来越受到研究者的重视，展现了广阔的应用前景【】。

设备指纹是指可用于唯一标识出该设备的设备特征或者独特的设备标识。目前较普遍的技术是针对设备提供的某些信息，从中提取特征生成近似唯一的设备指纹，与存储的可信信息匹配辨识，从而对设备的身份进行认证和辨识。其原理是不同厂商生产的设备（即便是同一厂商生产的设备）会由于硬件或软件的不一致在某些特定的方面存在些许差异。当前研究的技术手段可大致分为基于硬件的识别和基于软件的指纹识别。基于软件的识别通常是从流量信息中获取特征指纹，而基于硬件的识别则往往是根据设备的传感器信息或时钟偏移等硬件设备的信息对设备进行标识。

1.2.1 基于软件的设备识别

802.11 MAC帧的格式和内容经常被用于辨识无线设备。MAC地址是Medium/Media Access Control地址的简称，表示互联网上每一个站点的标识符。在网络底层的物理传输过程中，是通过物理地址来识别主机的，所以MAC地址就如同居民身份证上的身份证号码一样，具有全球唯一性。

Guo等人根据MAC帧头的信息追踪设备【Do You Hear What I Hear? Fingerprinting Smart Devices Through Embedded Acoustic Components，参考文献40】。顾杨等人利用从MAC层中的管理帧提取的特征指纹作为区分真假AP的特征指纹【】。他们将特征指纹又分为基于无线端的特征指纹和基于特种MAC帧刺激响应的特征指纹，其中基于无线端的特征指纹主要从信标帧和探测响应帧中提取，而基于特种MAC帧刺激相应的特征指纹主要从探测响应帧、关联响应帧和认证帧中提取。第一种特征指纹是通过被动的扫描无线网络获得的，第二种特征指纹是利用构造特殊格式MAC帧主动刺激AP（或无线路由器），捕获AP（或无线路由器）响应获得的。实验结果表明这两类无线设备的特征指纹可以检测出目前大部分无线钓鱼AP。但是，MAC地址是与网卡对应的，如果一个设备拥有不止一个网卡，这个设备就会有相应数目的MAC地址，此时MAC就不能用来唯一地标识设备；另外，如同居民的身份证容易被人盗用或者伪装，设备的MAC地址也容易被篡改或者伪装。

鉴于使用MAC的识别技术中存在上述缺陷，研究者们提出了一系列根据其他特征构建指纹的方法。Desmond等人仅通过研究802.11的请求探测帧，即可识别出连接在同一无线接入点（WLAN:）上的设备。【Do You Hear What I Hear? Fingerprinting Smart Devices Through Embedded Acoustic Components ，参考文献30】。Loh等人通过分析无线局域网（WLAN：Wireless Local Area Networks）中不同设备对802.11请求探测帧响应的时序特性，从中提取相应的指纹【Identifying Unique Devices through Wireless Fingerprinting】。Pang等人也能从流量中提取出特征指纹，进行设备的认证与标识【Do You Hear What I Hear，参考文献57】。Seika【Active Fingerprinting of 802.11 Devices by Timing Analysis】等人向802.11设备发送一个信号，捕获响应帧的到达时间并用时序分析研究其规律，用SVM分类器建立目标设备模型。~~他们在5台设备上测试该方法并达到86%的平均精度~~。

Gao K等人【A passive approach to wireless device fingerprinting】提出了一种用于确定接入到网络的AP的类型的基于黑盒的被动式识别技术。作者进行了大量的实验（收集超过60GB的数据）以对6种AP进行分类，在至少100000个数据包的数据基础上取得了较高的分类精度。无线网络中对AP的识别通常利用网络协议中常见的标识符，例如网络名称、MAC地址或者IP地址，但是这些标识符很容易被伪造、拦截或者更改。

此外，一些开源的工具如Nmap【Do You Hear What I Hear，50】和Xprobe【Do You Hear What I Hear，68】可以通过TCP/IP协议栈的响应识别设备的操作系统。其他基于软件的指纹识别技术则是根据设备上的应用如浏览器来确定设备。基于浏览器的智能终端识别的研究主要集中在桌面浏览器上。

Eckersley在2010年最早研究了浏览器指纹，他手机了涵盖物理层、应用层和用户层的浏览器特征【Android设备指纹识别技术，参考文献25】。他通过分析大量数据证实了浏览器指纹的有效性，同时也指出无线移动端设备（手机、ipad等）的浏览器识别较为困难。

在Eckersley之后，也有许多关于桌面浏览器识别的研究。Yen等人【同上69】曾经分析长达一个月的必应和Hotmail的日志，成功追踪到用户。Mowery等人通过JavaScript引擎执行Benchmark的结果对浏览器版本和操作系统等信息进行确认，同时通过对HTML5中引入的新元素<canvas>的分析发现，不同浏览器渲染绘制的图片会有所不同，可以形成浏览器指纹【Android设备指纹识别技术，参考文献26】。Acar等人通过研究JavaScript和一些较流行的第三方插件如Flash Player来获取设备上的字体列表，以此追踪用户【同上，18】。其他研究人员提出使用性能基准来区分JavaScript引擎【Fingerprinting Information in JavaScript Implementations】，浏览器的浏览历史也被用于追踪web端的用户【同上，56】。

1.2.2 基于硬件的设备识别

Bratus等人【Active behavioral fingerprinting of wireless devices】提出了一种主动式设备指纹识别方法,该方法通过向802.11无线设备发送一系列的经过特殊构造的某种错误格式的数据帧来观察设备的回应，通过这些回应的差异可以区分设备在芯片、固件或者驱动上的差异，以此达到设备识别的目的。同时，该方法也可以借助无线网卡芯片或无线网卡驱动的特征指纹信息来区分无线钓鱼AP和合法授权AP，为以较低成本来部署主动式无线端的检测方法提供了一些思路。

Radhakrishnan等人设计了一种名为GTID的识别框架，该技术采用主动和被动结合的方式，不仅可识别物理设备，还可以判断出相应的设备类型【GTID：A Technique for Physical Device and Device Type Fingerprinting】。GTID实现的主要依据是设备之间的异构性，他们认为不同的设备的处理器、DMA控制器和时钟偏移等都存在差异，而这些差异可以被用于识别设备和设备类型。GTID通过被动地抓取网络流量，从中提取出相应的指纹特征并使用人工神经网络（ANN）的算法进行训练和测试，取得了良好的效果。他们的技术适用于各种网络协议，且不需要进行深度包检测，但是由于该技术依赖于细粒度包时间，在路由器和交换机的缓冲区有时间丢失，故该技术的还有待进一步的研究。

基于硬件的指纹识别技术依赖于某些稳定的特性。【Estimation and removal of clock skew from network delay measurements】中的研究表明网络设备往往具有稳定的时钟偏移，Kohno和Cristea等人的研究便是基于此，他们通过分析TCP和ICMP中微小的时钟偏移，从而构建设备的特征指纹【Remote Physical Device Fingerprinting.被动】【Fingerprinting Smartphones Remotely via ICMP Timestamps】。该技术无需对设备做任何修改，即便目标设备通过多种方式接入互联网，或是测量设备与目标设备相距数千英里，均能达到较好的识别结果。

然而，时钟偏移率很大程度上依赖于实验环境【Specific emitter identification (SEI) and classical parameter fusion technology】。【Clock Skew Based Remote Device Fingerprinting Demystified】中作者研究了基于时钟偏差的无线设备识别的局限性，他们利用无线接入点（AP：Access Point）在信标帧中定期发送的时间戳为依据进行识别，消除了测量设备对时钟偏移的影响。此外，他们还进行了大量的评估，以探讨不同接入点和测量装置之间时钟偏差的分布及稳定性，发现时钟偏差的波动仅为1ppm。该算法能够消除测量装置对实验结果的影响，使不同设备生成的指纹可比较且具有区分性。

与仅使用时钟偏斜作为特征指纹不同的是，Neumann C等人【An empirical study of passive 802.11 Device Fingerprinting】分别以传输速率、帧大小、介质访问时间、传输时间、包内间隔时间等特征作为特征指纹进行802.11设备指纹识别，并比较这些特征在设备识别中的性能差异，实验结果表明传输时间和包内间隔时间的性能要优于其他参数，为其他的设备指纹识别工作提供了很好的借鉴。

Franklin等人提出了一种被动式的指纹识别技术，通过检测数据链路层的流量来检测IEEE 802.11设备上运行的固件和设备驱动程序的差异【Passive Data Link Layer 802.11 Wireless Device Driver Fingerprinting】。作者指出不同的无线网卡在扫描无线网络时发出的探测帧会有所不同，因为IEEE 802.11协议中并没有规定扫描的算法，这主要取决于无线网卡的驱动程序。他们通过分析终端传输信标帧的间隔时间提取设备指纹。通常情况下一个无线网卡加入网络时只发送极少数的请求，因此想要获取足够的数据量需要大量的时间。

Gerdes等人提出了基于模拟信号的终端设备识别技术，最多仅需要分析25帧就能高效识别设备。这种指纹识别技术是基于设备的制造和硬件组件在数字领域创造的独特的信号特征。【Device Identification via Analog Signal Fingerprinting】。虽然实验结果较好，但他们的研究仅是针对有线设备而言，这种方法对无线网络设备是否可行还存在疑问。而且这种方法依赖于如模数转换器和数字取样示波器等昂贵的设备，因此实用性不高。

此外，还有许多人研究了利用设备的辐射测量量作为指纹的技术【Wireless device identification with radiometric signatures】【Device Fingerprinting to Enhance Wireless Security using Nonparametric Bayesian Method】【Securing Wireless Systems via Lower Layer Enforcements】【Improved radiometric identification of wireless devices using mimo transmission】。其原理是天线、功率放大器、ADC、DAC等硬件在生产过程中不可能完全相同，每个设备因此会具有一系列独一无二的辐射信号，如振幅、频率、相位等，从中可提取出特征指纹。由于这些辐射测量量在设备生产后无法改变，所以使用起来安全可靠。这种方法最大的局限性在于数据采集困难，只能在有限的范围内获取到设备的指纹，无法进行远程追踪或认证。

Anupam Das【Do You Hear What I Hear? Fingerprinting Smart Devices Through Embedded Acoustic Components】等人认为厂商在制造智能手机的麦克风和扬声器时存在不同程度的缺陷，通过分析不同智能手机上的麦克风和扬声器的声学特征可以从中提取到设备相应的指纹。他们的方法虽然在50个安卓手机上达到了98%的精度，但是必须在手机的麦克风或扬声器发声的时候才能进行测试，实现复杂且无法远程监控。

在论文【AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable】中，作者认为智能手机和平板电脑中的加速度计具有独特的指纹，可以用于设备识别。他们在25个安卓手机和两个平板电脑上验证这一方法，识别精度达到96%。这种方法虽然精度高，但是要求设备上必要要有加速度传感器，且需要某种形式的外部刺激/震动来捕获加速度计的数据，局限性较大。相比而言，本文的工作只需远程捕获网络流量数据，不会被用户发现，操作简单，可实现性高。

1.3 本文主要研究内容和组织结构



本文以无线网络安全分析为研究背景、以网络中最为基础的流量数据为研究对象，提出了一种基于网络流量特征的无线设备指纹识别技术。~~这种方法通过捕获无线设备的网络流量，从中提取出能反映设备身份的特征指纹，据此对设备进行识别和认证。~~我们对基于特征指纹分析的无线网络设备的识别原理进行了讨论，从无线数据帧的生成和传输角度分析流量与设备的内部硬件组成、控制算法等因素的相关性，设计了基于特征指纹的无线设备指纹识别方案。研究框架如图\*\*所示。该方案主要分为数据采集、数据预处理、特征指纹生成、无线设备的训练和测试以及识别结果的评估与分析五个部分。

我们搭建了一个小型无线局域网用于数据采集，采集的对象为通过wifi接入到互联网中的移动设备，如智能手机、个人PC、ipad和kindle等。最终捕获到\*台设备共\*\*条无线帧数据。

数据预处理阶段从每条TCP协议数据帧中提取间隔时间（IAT）、帧大小（FrameSize）和传输速率（transRate）三个参数。为了排除网络延迟、数据量纲不一致等因素对实验结果的影响，我们对每项参数的数据都进行了降噪和归一化处理。

本文提出了两种特征指纹生成方法：基于概率密度的特征指纹和基于多特征融合的特征指纹。基于概率密度的特征指纹提取方法通过对每个参数的数据按顺序分组，使得每组包含一定数量的样本（一组的样本量用group\_size表示，实验中为500），再将每组数据划分为若干个窗（窗口数用bin\_size表示，实验中为20），用落在每个窗口内的频率来近似概率，构成设备的特征指纹。考虑到一个特征也许并不能完全表征设备的身份属性，因此我们将三种独立的特征融合在一起构成融合的特征指纹。

在无线网络设备模型的构建与评估阶段，我们使用了机器学习中五种常见的分类算法：随机森林（RF）、人工神经网络（ANN）、支持向量机（SVM）、k最近邻法（KNN）和朴素贝叶斯（Naïve Bayes），并采用准确率（precision）、召回率（recall）和F-mature对各个分类器的性能进行评估。实验结果表明

最后我们讨论了每组样本量、窗大小、设备数目、分类算法的参数对识别结果的影响。

1.4 论文的组织结构

第一章首先介绍了本课题的研究背景及意义，明确了课题的研究方向，接着阐述本文的主要研究工作。

第二章介绍基于网络流量的无线设备识别原理，通过分析网络数据帧的生成和传输过程，讨论无线网络流量与设备的内部硬件组成、控制算法等因素的相关性，并介绍了wireShark抓取的数据帧格式。

第三章是本文的研究方案，首先介绍了本文用于研究的数据集，紧接着针对无线设备提取问题提出数据预处理和特征指纹生成方法，然后设计出无线设备认证模型的构建和评估方案，最后展示实验结果并对影响识别的因素进行了分析和评价。

第四章阐述无线设备识别原型系统的具体设计和实现，并对原型系统进行了功能测试。

第六章对本文的研究工作进行总结，并分析识别技术和方案的不足之处，最后是对未来工作的展望。

2. 基于网络流量的无线设备识别原理和研究基础

2.1无线数据帧与设备个体的相关性

****

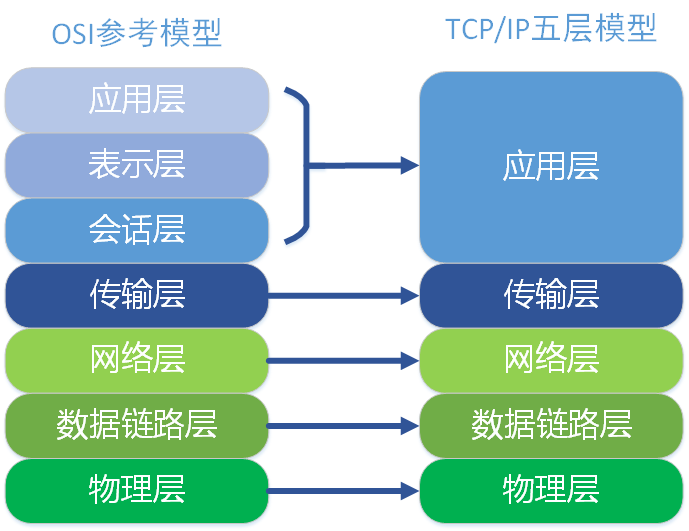
图 2‑1网络数据帧的生成过程

网络流量包的产生是一个复杂的过程，如图 2‑1所示，它需要设备内部多个硬件组件和操作系统的协作完成。这些硬件组件包括CPU、多层存储器结构（L1/L2 Cache，主存，硬盘）、DMA控制器、PCI总线、网卡等。流量包的产生从取值开始，首先从存储器结构中提取相应的指令集，发送到CPU进行执行；在操作系统的指示下，CPU创造一个或多个缓冲区描述符，缓冲区描述符包含起始的存储地址和流量需占用的存储长度，如果流量包在存储器中不连续地存储，这时CPU会产生多个缓冲区描述符，相应地，操作系统会指示CPU生成一个新的用于存储寄存器映射信息的缓冲区描述符到网卡中，这些信息需要经过前端总线、北桥和PCI总线发送到然后网卡会启动一个或多个DMA传输来检索描述符，之后网卡也会初始化一个或多个DMA传输把确切的流量数据帧从主存转移到网卡的传输缓冲区，这些数据由前端总线离开，经由北桥和PCI总线传输到网卡中。最后，网卡通知操作系统和CPU描述符已经处理完成，相应的流量包也产生成功发送到网络中。

网络流量包的产生过程，这个过程包括多个硬件组件和操作系统的协同工作，所以网络流量包中会包含能体现出发送设备身份的相关信息，比如设备所用的操作系统、设备CPU的配置、处理器所使用的主要算法、设备每个硬件的使用时限等。这种相关性不仅会体现在不同厂商、不同类型的设备中，也会体现在相同型号的设备中，因为即使是同一型号的设备，设备内部所采用的硬件部件、硬件部件的使用时间、处理器的时钟频率这些信息也不可能完全相同，犹如两个双胞胎也不会有完全相同的指纹特征和基因图谱。正是这种相关性在设备层次和组件层次上的差异为本文通过网络流量进行设备和设备类型的指纹识别提供了理论依据和一种全新的思路。

2.2 流量数据帧介绍

TCP/IP是Transmission Control Protocol/Internet Protocol的简写，中译名为传输控制协议/因特网互联协议，又名网络通讯协议，是Internet最基本的协议、Internet国际互联网络的基础，由网络层的IP协议和传输层的TCP协议组成。TCP/IP定义了电子设备如何连入因特网，以及数据如何在它们之间传输的标准。协议采用了4层的层级结构，每一层都呼叫它的下一层所提供的协议来完成自己的需求。TCP/IP参考模型与OSI（Open System Interconnection，开放式系统互联），[开放式系统互联](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E5%BC%8F%E7%B3%BB%E7%BB%9F%E4%BA%92%E8%81%94)七层参考模型的对比示意图如下。



在TCP/IP参考模型中，去掉了OSI参考模型的会话层和表示层（这两层的功能被合并到应用层实现），下面将介绍各层的主要功能，并结合wireshark捕获的数据介绍各层的数据格式。

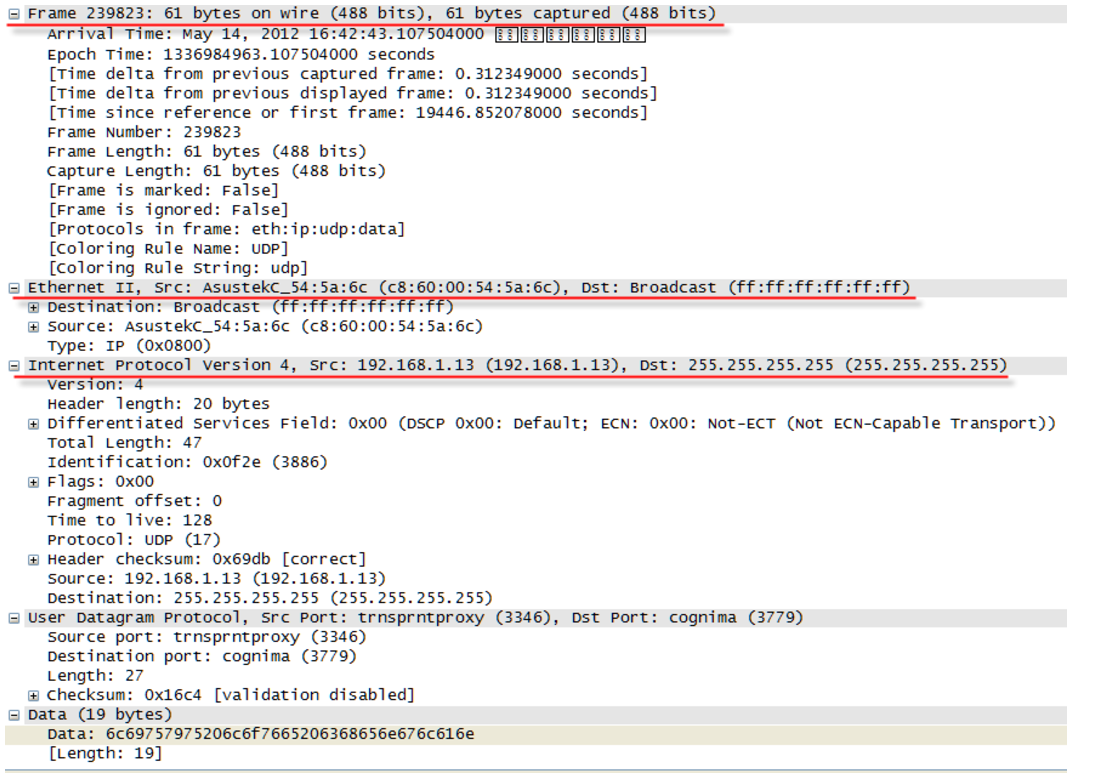


图 2‑2 流量数据帧结构示意

1.应用层

应用层是最高层，直接面向应用程序提供服务，其作用是在实现多个系统应用进程相互通信的同时，完成一系列业务处理所需的服务。应用层面向不同的网络应用引入了不同的应用层协议，其中，有基于TCP协议的，如文件传输协议（File Transfer Protocol，FTP）、虚拟终端协议（TELNET）、超文本链接协议（Hyper Text Transfer Protocol，HTTP），也有基于UDP协议的。

2.传输层

在TCP/IP模型中，传输层的功能是使源端主机和目标端主机上的对等实体可以进行会话。在传输层定义了两种服务质量不同的协议。即：传输控制协议TCP（transmission control protocol）和用户数据报协议UDP（user datagram protocol）。

TCP协议是一个面向连接的、可靠的协议。它将一台主机发出的字节流无差错地发往互联网上的其他主机。在发送端，它负责把上层传送下来的字节流分成报文段并传递给下层。在接收端，它负责把收到的报文进行重组后递交给上层。TCP协议还要处理端到端的流量控制，以避免缓慢接收的接收方没有足够的缓冲区接收发送方发送的大量数据。如图\*\*是wireshark捕获的数据帧，其传输层的TCP协议报文格式如下：

Transmission Control Protocol, Src Port: 2764 (2764), Dst Port: http (80), Seq: 0, Len: 0 传输控制协议TCP的内容;  
Source port: 2764 (2764）源端口名称（端口号）;  
Destination port: http (80) 目的端口名http（端口号80）;  
Sequence number: 0 (relative sequence number) 序列号（相对序列号）;  
Header length: 28 bytes 头部长度;  
Flags: 0x02 (SYN) TCP标记字段（本字段是SYN，是请求建立TCP连接）;  
Window size: 65535 流量控制的窗口大小;  
Checksum: 0xf73b [correct] TCP数据段的校验和;  
Options: (8 bytes) 可选项;

UDP协议是一个不可靠的、无连接协议，主要适用于不需要对报文进行排序和流量控制的场合。

3.网络层

网络层是整个TCP/IP协议栈的核心。它的功能是把分组发往目标网络或主机。同时，为了尽快地发送分组，可能需要沿不同的路径同时进行分组传递。因此，分组到达的顺序和发送的顺序可能不同，这就需要上层必须对分组进行排序。

网络互连层定义了分组格式和协议，即IP协议（Internet Protocol）。

4.数据链路层

在物理层提供比特流服务的基础上，建立相邻结点之间的数据链路，通过差错控制提供数据帧(Frame)在信道上无差错的传输，并进行各电路上的动作系列。数据链路层在不可靠的物理介质上提供可靠的传输。该层的作用包括：物理地址寻址、数据的成帧、流量控制、数据的检错、重发等。在这一层，数据的单位称为帧(frame)。数据链路层协议的代表包括：SDLC、HDLC、PPP、STP、帧中继等

5.物理层

物理层处于最底层，是整个计算机网络的基础。物理层为设备之间的数据通信提供传输媒体及互连设备，为数据传输提供可靠的环境。

由上述数据帧的结构可以看出，数据帧中包含了设备在访问互联网时的各种时空信息，为本文基于数据帧进行设备行为特征身份识别奠定了基础。

2.3 本章小结

本章介绍了网络数据帧的生成和传输过程，并从该角度分析了数据帧与无线设备个体之间的相关性，从原理上说明了基于网络流量的设备识别技术的可行性。紧接着介绍了TCP/IP参考模型和wireShark抓取的TCP数据帧格式，为后文的研究提供基础。

3无线设备识别研究方案

本章先介绍实验的软件环境和硬件环境。接着详细阐述了无线设备指纹识别研究方案中各个环节使用的方法和关键技术，包括数据采集、数据预处理和特整提取。接下来在现有数据集上验证了技术的有效性。最后对影响识别的因素进行进一步的分析和讨论。

3.1 实验环境

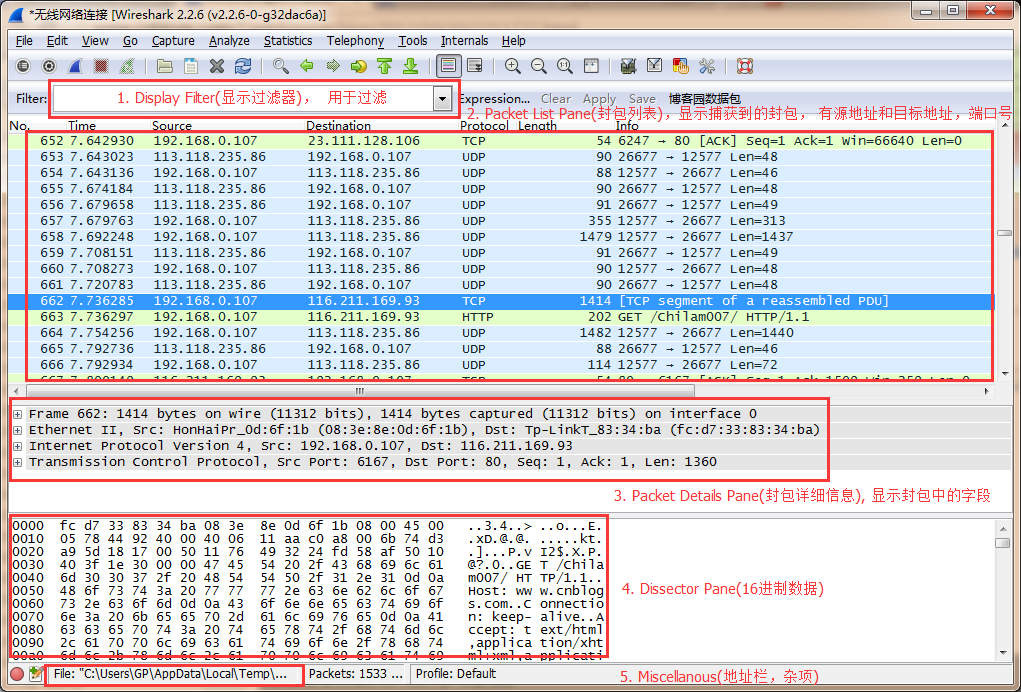
3.1.1 软件环境

（a）抓包工具介绍

WireShark是一款常见的网络数据包分析工具。WireShark具有以下特性：

1. 支持UNIX和Windows平台；
2. 可以在线截取各种网络封包，显示网络封包的详细信息，也可以分析已有的报文数据，包括http、TCP、UDP等网络协议包；
3. 提供多种过滤规则，进行报文过滤；
4. 可进行多种统计分析；

需要注意的是WireShark只能查看封包，而不能修改封包，也无法发送封包。我们在实验中用wireshark进行网络流量包的抓取，其主界面如图【】所示。抓取完流量后，在过滤框内输入“ip.src == 192.168.1.110 and TCP”，通过“导出特定分组”便可将无线终端设备发出的TCP流量以文件的形式存储下来，文件格式为pcapng。



（b）jnetPcap介绍

jnetPcap是一个开源的java类库，主要功能是捕获和分析数据包。因为Java平台本身不支持底层网络操作，需要第三方包利用JNI封装不同系统的C库来提供Java的上层接口，jNetPcap是[libpcap](http://jnetpcap.com/)的一个Java完整封装，jNetPcap使用与libpcap相同风格的API。libpcap是unix/linux平台下的网络数据包捕获函数库，大多数网络监控软件都以它为基础。jnetPcap主要有下面几个特点：

1. 提供几乎所有libpcap类的封装；
2. 可实时解码所捕获的数据包；
3. 提供广泛的网络协议库（核心协议）；
4. 用户可以使用java SDK轻松添加自己的协议定义；
5. jnetPcap可以使用本机和java混合实现最佳的分组解码性能。

由于我们的实验是在Windows系统下进行的，使用jnetPcap还必须要安装WinPcap，以提供jnetPcap所需要的链接库。Pcap类是jnetPcap中最为核心的类，是一个对libpcap中方法的Java直接映射，提供了获取网卡设备列表，设置过滤器、数据包分析等必须的工作。

虽然wireShark也具有流量数据包解析的功能，但只能手动打开每一条流量数据查看详细信息，我们实验中对每一台无线设备都抓取了上万条流量，使用wireShark解析数据并提取特征是一项不可能完成的任务。本文利用jnetPcap开源库对捕获的网络流量进行解析，并提取出每个设备构建特征指纹所需要的参数。

（c）开发语言介绍

由于jnetPcap开源库是基于Java平台的，因此本文使用Java语言解析网络流量，并从中提取构建设备特征指纹所需要的参数。除此之外，本文使用python语言进行特征指纹的构建、数据的分析和处理，基于分类算法的设备身份模型的构建和验证。

3.1.2 硬件环境

实验中搭建无线网络环境用到的硬件设备主要有一台笔记本电脑，两个路由器和一个镜像交换机，其具体信息如表【】所示：

|  |  |  |
| --- | --- | --- |
| **名称** | **主要参数** | **描述** |
| Dell笔记本 | 型号：Ispiron 灵越15 5000系列  CPU主频：2.5GHz  内存：4GB | 装有Windows10系统，WireShark软件，用于网络监控，捕获数据流量 |
| TP-LINk路由器 | 型号：TL-WR842N  无线传输速率：300Mbps  无线频段：2.4GHz | 支持IEEE 802.11b/g/n，具有良好的无线性能和连接稳定性，用于无线帧的发送与接收 |
| HUAWEI路由器 | 型号：WS550  最高传输速率：450Mbps  无线频段：2.4GHz | 支持IEEE 802.11n，兼容IEEE 802.11b/g用于建立Wifi热点 |
| TP-LINK端口镜像交换机 | 型号：TL-SF2005  5个10/100Mbps RJ45端口 | 支持端口镜像功能，提供一个固定上联端口，一个固定监控端口和三个普通端口，用于监控网络流量 |

3.2 数据采集

3.2.1无线网络环境

无线网络环境下的数据采集是在西安交通大学校园内完成的。无线网络是指利用无线通信技术搭建的网络，通常分为借助公众移动通信网实现的无线网络（如4G、3G或GPRS）和无线局域网（WiFi）两种方式。由于借助公众移动通信网实现的无线网络环境下的数据采集需要运营商的配合，数据获取困难，因此我们在实验中采用的是第二种方式，即自己设置WiFi热点，搭建无线局域网。

为了探讨移动设备在远程网络下的流量行为，同时研究流量数据帧在网络中的传输是否会对识别造成影响，我们在实验中搭建了一个具有两个节点路由器的无线网络，其设计模型如图【】所示。



该网络包含一个镜像交换机、两个路由器和若干个无线终端设备。一般情况下，与交换机或者路由器相连的电脑只能够接收到自己的数据包，为了网络管理的需要，有些交换机或者路由器提供了端口镜像的功能。端口镜像就是把交换机的源端口的流量完全拷贝一份，从目的端口发出去，这样既可以分析网络流量，同时不影响原来的数据发送。有了镜像交换机，即可通过网络分析软件监控整个网络的运行状况和上网数据。

两个路由器一个与镜像交换机相连，一个用于开启WiFi热点，建立小型局域网。移动终端连接到这个局域网后，路由器会跟配一个临时IP地址给该终端设备（例如局域网网关为192.168.1.1/24，设备被分配的IP地址将会是192.168.1.X），但是设备被分配的这个IP地址仅在这个局域网内有效，流量数据帧经由路由器中转之后，数据帧中的IP地址便会转变为路由器WAN口（?）的IP地址。假设路由器WAN口的IP地址为202.117.14.191，那么设备发出的数据帧经过终端路由器转发后数据帧的源IP就会由局域网内的192.168.1.X变为202.117.14.191。在这种情况下，局域网内所有移动设备的IP地址都会表现为202.117.14.191，为了在设备数据训练与测试过程中给样本数据进行标记，我们在采集数据的过程中独立地采集每台设备的流量数据，即在一次采集过程中只有一台移动设备接入局域网中，这样流量数据帧源IP为202.117.14.191都认为是移动终端发出的数据。

真实的采集环境如图【】所示…. 图的左边是可以是表3-1中十台移动设备中的任一台，它连接入终端路由器的局域网之后，发出的流量便可以通过右边的网络传输到两跳之后的镜像交换机。然后在PC上通过调用jnetPcap类库实现的java程序便可以通过镜像交换机的镜像口抓取到测试设备发出的流量。在java程序中，可以设置对于流量类型和源IP等过滤条件对原始流量数据进行过滤，功能与在Wireshark上设置“ip.src == 192.168.1.110 and TCP”的过滤条件等同，通过“导出特定分组”便可以将数据流量存为由测试设备发出的流量类型为TCP的PCAP文件。

其中192.168.1.110为中间路由器WAN口的IP地址，由于终端路由器的局域网中一次仅有一台测试设备接入，而路由器在稳定的网络环境下不会主动发送TCP流量，因此可以认为这个无线网络中的TCP流量都是由局域网中的移动设备发出的，可以将此数据视为测试设备的训练数据。

3.2.2 数据采集方案

由于WireShark采集数据。。。。，所以每台设备数据的采集都是单独进行的，即一台设备采集结束，数据保存后再进行下一台设备的采集。同时由于不同的应用服务程序用的协议不同，数据采集的方案也不相同，下面详细介绍每一种协议流量捕获的方案。

1. TCP数据采集方案

TCP（Transmission Control Protocol，[传输控制协议](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)）是面向连接的、可靠的传输层通信协议，许多需要高度可靠的面向连接的服务都使用了TCP，移动终端上的大部分应用程序都是基于TCP协议的，因此用户只需要使用手机的网络应用程序即可捕获到TCP数据流量。但是诸如微信、淘宝等应用程序产生的TCP的流量较少，采集足够的数据所花费的时间过长，因此我们让每个终端设备播放视频，可在短时间内产生大量的流量。

数据采集过程如下：第一步，无线终端接入WiFi热点，设置好WireShark，准备捕获流量；第二步，无线终端播放任意视频，直到捕获到了足够量的数据；第三步，设置过滤规则，保存数据流量。

每个无线终端大约需要**8w条**TCP数据，大约持续2到3小时即可捕获到足够的数据。

1. UDP数据采集方案

UDP（User Datagram Protocol，用户数据报协议）是[OSI](https://baike.baidu.com/item/OSI)参考模型中一种无连接的[传输层](https://baike.baidu.com/item/%E4%BC%A0%E8%BE%93%E5%B1%82)协议，提供面向事务的简单不可靠信息传送服务。UDP与TCP协议一样，用于处理数据包，位于OSI模型的第四层——传输层，在IP协议的上一层。。UDP用来支持那些需要在[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)之间传输数据的网络应用，包括[网络视频会议](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E8%A7%86%E9%A2%91%E4%BC%9A%E8%AE%AE)系统在内的众多的客户/服务器模式的网络应用都需要使用UDP协议。因此无线终端在播放视频时也可以采集到UDP数据包（帧？）。

UDP数据采集方案与TCP相同，播放视频2到3小时大约可产生？？条数据。

3.2.3 数据集介绍

我们用于实验的移动终端有PC、智能手机、平板电脑和亚马逊kindle等十余台设备，它们的硬件及软件配置信息如表【】所示。数据采集时间为2017年3月25日至2017年4月18日，在这\*天中\*台设备的采集工作独立进行，每台设备的数据采集规模都达到了600MB以上，数据帧数目至少100万条，持续时间在两个小时到四个小时之间，具体时间视数据采集时的网络状况而定。（每个设备的数据规模可绘制柱状图，同样的，在解释处理完的数据时，也可以画柱状图）

|  |  |  |  |
| --- | --- | --- | --- |
| 品牌 | 操作系统 | CPU | RAM |
| 小米3 | Android 4.4 | Quad-core 2.3GHz | 2GB |
| 小米4 | Android 6.0 | Quad-core Max  2.5GHz | 2GB |
| 小米5S | Android 6.0 | Quad-core Max 2.15GHz | 4GB |
| iphone5 | ios 6.0 | 苹果A6 1.0GHz | 1GB |
| 华为 honor7 | Android 5.0 | Hisilicon Kirin 935 | 3GB |
| 魅蓝 note | Android 4.4 | Flyme0S4.2.0.4A | 2GB |
| DELL 7420 | Windows7 | Intel Corei5-3230M 2.6GHz | 8GB |
| DELL Vostro 3550 | Windows7 | Intel Corei5-2410M 2.3GHz | 4GB |
| Thinkpad X240 | Windows10 | Intel Corei5-4200U 2.29GHz | 4GB |
| ipad Air | Ios8.4.1 | 64 位 A7 芯片 /M7 处理器 | 1GB |

3.3 数据预处理

数据预处理模块的功能是对捕获的流量帧数据进行解析，从中提取出构建特征指纹所需参数，并对这些参数进行降噪和归一化处理，将其转化为便于提取特征的形式。

数据预处理的流程如图【】所示。先打开pcap文件，对每一条数据帧提取帧号、帧间隔时间、帧大小和传输速率这四项参数；对除帧号外的参数进行降噪处理；再将降噪后的数据归一化，处理完成后保存结果。其中，解析pcap文件并提取参数使用的是java语言，数据降噪和归一化处理用的是python语言。下面逐一介绍这几项操作的详细过程。



（1）数据帧参数提取

对pacp文件的处理需要用到java的jnetpcap库，使用jnetpcap类库提供的接口打开pcap文件，对其中每一条数据帧记录，从帧头提取如下几个参数：

1. **帧号（Frame No）**：每一个数据帧在文件中的编号，它们有可能是不连续的，这一参数主要是用于计算帧间隔时间。
2. **帧间隔时间（IAT：Interval-arrival time）**：两帧到达时间的间隔，由于我们用于实验的数据帧不一定是连续的，单纯用当前帧的到达时间减去上一帧的到达时间来计算帧间隔时间是不合理的。我们采用的做法是用两帧到达时间之差除以两帧的帧号之差，计算公式如下：

其中：t1和t2分别为两帧的到达时间，F1和F2分别为两帧的帧号。

1. **帧大小（FrameSize）：**每一帧的大小。
2. **传输速率（TransRate）：**网卡进行数据传输的速率，单位为Mbps（兆位/秒），反映了网卡每秒钟接受或者发送数据的能力，计算公式为：

TransRate = FrameSize / IAT

为了保证每个设备的数据量一样多，方便进行后期的训练和测试，我们从每个协议的数据中按顺序选取N条数据帧提取参数，多余的弃之不用。对TCP协议，N=80000；对UDP协议：N=。。。。经过上述预处理，得到的每个设备每个协议的数据如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 帧号 | 帧间隔时间 | 帧大小 | 传输速率 |
| 1 | 0.038666 | 130 | 3362.126933 |
| 2 | 0.018444 | 54 | 2927.781392 |
| 5 | 0.057298 | 60 | 1047.156969 |
| … | … | … | … |
| N | 0.061937 | 60 | 968.726286 |

1. 数据降噪

由于网络环境的复杂性，网络数据的传输会受到网络带宽和网络延时等因素的影响，例如帧间隔时间就会受到数据包延迟的影响而变得异常大。这种不和其他数据相一致的数据我们称之为噪声数据。噪声数据会干扰设备识别的结果，需要对其进行处理。常见的噪声数据处理方法有：分箱、聚类、计算机和人工检查结合和回归，我们在实验中采用的是自定义区间的方法，属于分箱降噪的一种。

自定义区间降噪的做法是：用户根据需要自定义区间，只查看特定范围内的数据。这种方法的关键在于阈值的选取，只要选出合适的阈值，即可查看在阈值范围内的正常数据。

为了选出合适的阈值，我们绘制出每个参数的概率密度（PDF：probability density function）曲线。PDF曲线反映了随机变量在某个确定取值点附近可能性的大小，而随机变量的取值落在某个区间内的概率则为概率密度曲线在该区间上的面积。如果随机变量在某一点附近的概率密度值过小，则说明该范围内的数据量非常少，即为我们所提到的异常数据，也就是噪声数据。

以TransRate为例，随机选取一台无线设备，绘制其PDF曲线如图【】所示。图中横轴代表IAT取值，纵坐标代表在某一点上取值的概率。曲线呈非常明显的脉冲状，说明IAT取值落在某一区间内的数据量非常大；同时在其余部分的概率密度值无限接近于零，说明该范围内的数据很少，可以将其作为噪声数据滤除。通过多次实验和对多个设备数据的观察，我们发现以1.2\*106为阈值时能过滤掉大部分的异常数据，数据过滤后的PDF曲线如图【】所示。





通过分析，三个参数的阈值分别为：IAT=0.4， FrameSize= 2000， transrate=1.2\*10^6 ：。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 帧间隔时间 | 帧大小 | 传输速率 |
| 上限阈值 | 0.038666 | 130 | 3362.126933 |
| 下限阈值 | 0.018444 | 54 | 2927.781392 |

1. 数据归一化

不同属性的数据往往具有不同的量纲或量级，如IAT的取值约在\*\*到\*\*之间，而传输速率的取值则在\*\*到\*\*之间。为了防止量级影响数据分析的结果，消除指标之间的量级影响，需要对数据进行归一化处理，这对后续的处理十分必要，很有可能会提高实验精度。

数据的归一化是将数据按比例缩放，使之落入一个小的特定区间。由于信用指标体系的各个指标度量单位是不同的，为了能够将指标参与评价计算，需要对指标进行规范化处理，通过函数变换将其数值映射到某个数值区间。常用的方法有min-max标准化、z-score 0均值标准化和非线性归一化。

本文使用max-min归一化方法，计算公式为：

其中：xmin为某个属性的最小值，xmax为某个属性数据的最大值，x表示待归一化处理的数据，y表示归一化处理之后的数据。

从公式中可以看出，Min-max标准化的结果是将数据向量落在[0,1]区间内。

2.4 本章小结

本章介绍了数据采集的环境，通过搭建一个小型的无线局域网，用wireshark捕获连接到网络中的设备流量。为了融合多种协议中包含的设备身份信息，以加强设备识别的效果，本文共采集了TCP、UDP、、、这几种协议的数据流量。同时由于不同应用服务程序依赖的协议不同，针对上述每种的协议我们设计了相应的数据采集方案：通过播放视频捕获TCP和UDP流量；通过脚本程序让设备自动发送邮件以捕获\*\*\*协议。最终共采集到\*\*个设备的流量数据，每个设备采集到N条TCP协议数据帧、N条UDP协议数据帧\*\*\*\*。之后解析这些数据，从每条数据帧中提取帧间隔时间、帧大小和传输速率三个属性的数据，并对其进行降噪和归一化处理，将数据转换成易于提取特征的形式，为无线设备指纹识别提供数据集。

3.4 特征指纹生成

（1）PDF度量

为了初步度量提取的各项参数能否体现移动设备在硬件组成和控制算法上的差异，我们绘制出同一参数在不同设备上的PDF曲线。通过观察PDF曲线的分布情况，可以大致判断该参数是够能反映出设备之间的差异。如果不同设备某个参数的PDF曲线有较为明显的差异，我们可以认为不同设备在该参数上的分布不同，也意味着该参数可以作为设备识别的依据。

以帧时间间隔（IAT）为例，我们随机挑选了三台无线设备（cyf，jzp，wz），分别从其流量数据中提取帧时间间隔信息，并绘制出PDF曲线，如图\*\*所示。从图上可以看出，三条曲线在IAT取值较小时完全不重合，因此我们有理由相信，帧时间间隔对于移动设备具有很好的区分性。同样的，包大小和传输速率在无线设备识别中也表现出了良好的性能。



3.4.1 基于概率密度的特征指纹

由于原始数据量过大，直接将原始数据作为设备的特征指纹输入到分类器中是不可取的。通过3.1.1（PDF分析）中对各项的参数的分析中可以看出，每个参数的概率密度曲线可以很好地区分不同的设备。基于这个思路，我们提出基于概率密度的特征指纹构建方法。由于每个参数服从的概率密度函数未知，在样本量较大的前提下，可以用频率近似替代概率。

图 特征指纹的生成过程

特征指纹的生成过程如图【】所示。为了详细解释此过程，不妨假设每个参数的原始数据有N条，基于概率密度的特征指纹构建操作步骤如下：

第一步：设这N条数据为x1，x2，…,xn,将他们按顺序划分成大小相同的若干组（group），设定每组的样本量（group\_size）为M，即每组有M条数据。原始数据共被划分成T=N/M个分组，即group\_1,group\_2,group\_T，对第i（1<i<T）组数据，该组数据为[xm\*(i-1)+1,…,xM\*i],划分情况如表\*\*所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 组号 |  |  |  |  |
| Group\_1 | X1 | X2 | … | XM |
| Group\_2 | Xm+1 | Xm+2 |  | Xm+m |
| … | … | … |  | … |
| Group\_T | Xm\*(t-1)+1 | Xm\*(t-1)+2 |  | Xm\*t |

第二步：将每组数据划分为W个窗口，统计该组数据落在每个窗口的频率，形成1\*W维的特征向量：

 (3-3)

其中值样本数据落在每个窗口内的频率。

第三步：将各组数据得到的特征向量组合在一起，构成特征矩阵（T个组\*w个窗口），这个矩阵即为设备的特征指纹，如表【】所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 帧号 | 帧间隔时间 | 帧大小 |  | 传输速率 |
| Group1\_F | Group1\_Freq1 | Group1\_freq2 | … | Group1\_freqw |
| F2 | Xm+1 | Xm+2 |  | Xm+m |
| … | … | … |  | … |
| Group\_T | Xm\*(t-1)+1 | Xm\*(t-1)+2 |  | Xm\*t |

其中group\_i\_F(1<i<w)表示从第i组数据中提取的特征向量。

实验中，我们设定每组的数据量M=300，窗口大小W=20。由于我们从每个设备的流量数据中提取出三个参数（frame size，IAT， TransRate），因此按照上述操作步骤造作，将会形成三个t\*w维的特征矩阵，每个特征矩阵都可以作为设备的特征指纹来验证设备的身份。

3.4.2 基于特征融合的特征指纹

考虑到从一个参数中提取的特征指纹也许并不能完全表征设备的身份属性，独立的特征可能仅能表征设备在一个方面的属性，因此我们考虑将三种独立的特征融合在一起生成特征指纹。在式\*\*的基础上，将从三个参数中提取出的特征向量合成，便形成了一个1\*（w\*3）维的特征向量具体如\*\*所示：

 (3-4)

其中指帧间隔时间在样本内的概率分布，指帧大小在样本内的概率分布，指传输速率在样本内的概率分布。

最后将所有的向量，形成T\*（w\*3）维的特征矩阵，即为该设备基于特征融合的特征指纹。

3.5 指纹识别模型的构建与评估

3.5.1 分类器介绍

3.5.1.1 随机森林

随机森林（RF：Random Forests）是一种比较新的机器学习模型。上世纪八十年代Breiman等人提出分类与回归树的算法（CART：Classification and regression tree）【Breiman L, Friedman J H, Olshen R A, et al. Classification and Regression Trees[M]// Classification and regression trees. Wadsworth International Group, 1984:17–23.】，通过反复二分数据进行分类或回归，将输入空间即特征空间划分为有限个单元，并在这些单元上确定预测的概率分布。

2001年Breiman又将分类树组合成随机森林【Breiman L. Random Forests[J]. Machine Learning, 2001, 45(1):5-32.】。随机森林由很多的决策树组成的，这些决策树之间互相没有关联。形成森林之后，当有一个新的输入样本进入的时候，每个决策树测试待分类项中相应的特征属性（这些属性可以是分类项不同的属性），并按照判定条件将输入样本归类到决策树不同的分支，直到最后到达叶节点，叶节点即为决策树预测的类。最后看哪个类被选择的最多，就预测这个样本属于哪个类。如图2‑4，要对一个人的身份进行二分类判定，不同的决策树就可以分别根据该人的年龄、是否是学生、信用评级等身份属性来判定，最后根据投票结果进行投票表决，示例中，五个叶节点，三个判定为“是”，两个判定为“不是”，所以对这个人的身份作肯定的判定。

随机森林的“随机”体现在以下两个方面：1）在训练每棵树时，从训练样本池（N个样本）中选取的训练子集（n个样本，n<N）是随机有放回地选取的。这样在训练时，每一棵树的输入样本都不是全部的样本，可以避免过拟合现象；2）在决策树的每个节点进行分裂时，从输入特征（M个特征）中随机选取m个输入特征（m<M）,然后从这m个特征里选一个最好的进行分裂。与分类回归树算法相比，随机森林在运算量没有显著增加的前提下提高了预测精度。



图 2‑4随机森林示例

随机森林的优点很多。1）与其他分类器比较，随机森林在很多数据集上，都取得了很好的分类性能，表现良好；2）随机森林训练中可以随机选择特征子集，因此它在处理高维度（即多特征）的数据时，不需额外进行特征选择；2）随机森林对数据训练完之后，能够给出对不同特征的分析结果，比如哪些特征比较重要；3）随机森林的时候使用无偏估计处理泛化误差，泛化能力较其他模型更强；4）随机森林的不同决策树之间是相互独立的，对决策树并行化训练，训练速度快；5）更适合处理不平衡的数据集，可以平衡不同数据集之间的误差；6）如果数据集中有大部分的特征遗失，仍可以保持较高的准确度。但是随机森林也有一定的缺点，比如在某些噪声较大的分类或回归问题上随机森林会产生过拟合。

3.5.1.2 人工神经网络

ANN是启发于生物学上的神经网络的计算模型，从结构和功能上都与生物神经网络类似。一个ANN由一组名为神经元的互联计算模型所组成，这些神经元根据特定的激活函数将输入转换为输出。虽然在我们的工作中仅将ANN视为一个黑盒，但是我们主要使用由一个输入层、一个隐含层和一个输出层组成的前馈神经网络。该类ANN通常需要监督式学习用于预测、模式识别和非线性函数拟合。在我们的ANN中，使用共轭梯度反向传播作为训练函数，ANN输出0-1之间的值，其中1代表完美匹配。

图 2‑7显示了一个ANN的示意图，这个ANN可使用含有N个窗口的指纹从而将M个不同的设备或设备类型进行分类，这是一个包含一个输入层、一个隐含层和一个输出层的多层前馈ANN。输入层接受大小为N的向量（）,并产生一个大小为M的输出向量（）。输入向量的元素对应于概率密度（签名）中的值，并且输出向量的元素对应于输入签名与被训练的M个设备或设备类型签名之间的相似性度量。

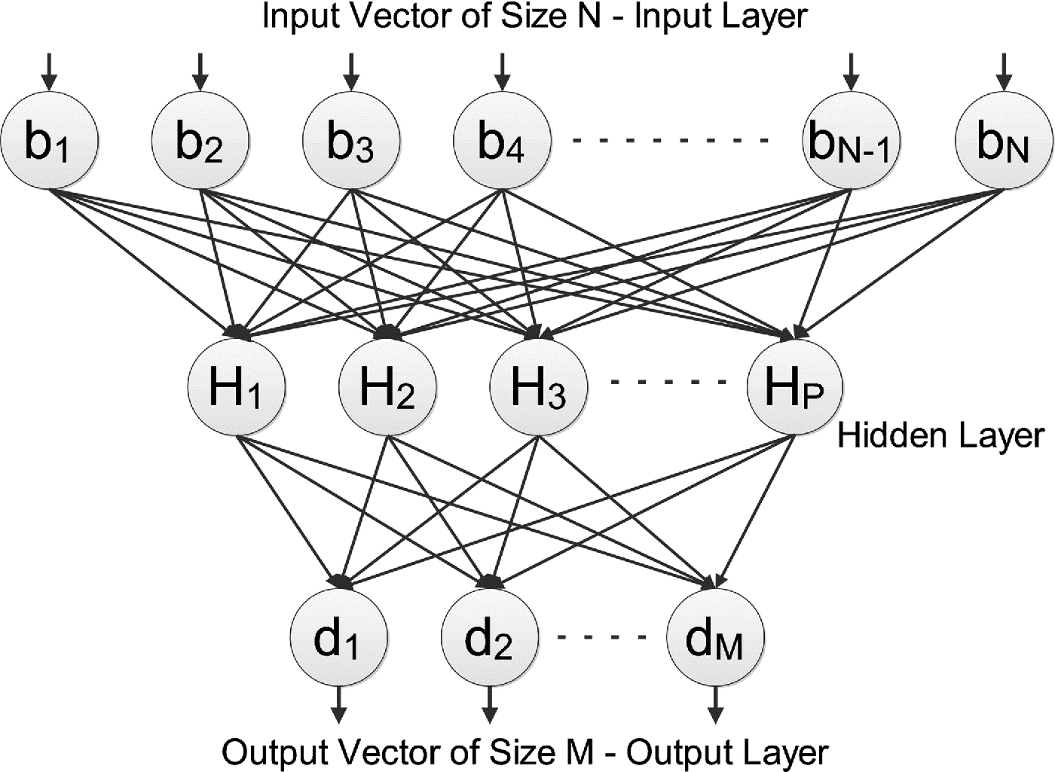


图 2‑7简单的神经网络示例

图2所示输入层、隐层1、隐层2和输出层的任意节点分别用，，和表示，彼此之间的权重系数分别用，，表示。输入层节点的输入用表示，隐层和输出层节点的输入用表示，激励输出用表示，则隐层1第个神经元的输入和激励输出可分别表示为、。

设训练集为，任意训练样本为，经过正向输出对应的实际输出为，期望输出为。迭代次数为。则双隐层BP算法的学习过程如下：

正向传播：输入层到隐层再至输出层的逐层传播。

对输入层任意训练样本，隐层1，隐层2，输出层节点的输入和输出依次为：

，，；(1)

，，；(2)

，，；(3)

，(4)

其中激活函数均采用Sigmoid函数：

。(5)

则根据以上计算得到输出层第个节点的误差信号，

。(6)

由于为双隐层网络，则网络上的均方误差为

。(7)

误差逆向传播：输出层经隐层反向传播至输入层，逐步调整连接权值。

在迭代的每一轮中，BP神经网络基于LM算法对参数进行调整。以下为输出层与隐层2，隐层2与隐层1，隐层1与输出层之间的连接权值更新公式计算过程：对式(7)的误差，给定学习率（取），有

；(8)

；(9)

。(10)

其中表示Jacobian矩阵，为单位矩阵。

注意到先影响到第个输出层节点的输入值，再影响到其输出值，然后影响到误差，所以有

，(11)

根据的定义式(3)，显然有

，(12)

又根据激励函数的性质，即

，(13)

则根据式(4)(6)(7)，有

，(14)

所以最终得出

。(15)

同理可推得：

，(16)

。(17)

引入动量因子（取）对学习率进行调整，减小震荡，提高网络的训练速度。得到改进的修正连接权值量分别为：

，

，

。(18)

所以下一次迭代时新的修正连接权值为：

，

，

。(19)

人工神经网络的分类准确度较高，并行分布处理、分布存储及学习能力强，对噪声数据有较强的健壮性和容错能力，能较好地处理复杂的非线性数据。但是，人工神经网络的实现需要较多的参数设置，比如网络拓扑结构、阈值的选择等；而且神经网络训练对于计算机资源的消耗较大，训练所需的时间复杂度较高。

3.5.1.3支持向量机

支持向量机（SVM: Support Vector Machine）是Cortes和Vapnik在1995年提出的经典分类算法【Cortes C, Vapnik V. Support-vector networks[J]. Machine Learning】，由于在解决小样本、非线性及高维模式识别中表现出许多特有的优势【小贾毕设，15】，很快成为机器学习的主流技术，并直接掀起了“统计学习”（statistical learning）在2000年前后的高潮【周志华. 《机器学习》[J]. 中国民商, 2016(3).】。

SVM是一种有监督式学习的分类模型，它分类的思想是：给定一个包含正例和反例的训练样本集合，SVM找到一个超平面对样本根据正例和反例进行分割（如图2-5），超平面应当位于正例和反例之间且距离两类数据样本的距离尽可能远。SVM算法的核心在于找到一个核函数，使得在低维空间中线性不可分的问题在高维空间中变得线性可分，这样做的好处是降低在原数据空间进行非线性曲面分割计算的复杂度。

选择一个恰当的核函数不仅可以在很大程度上降低SVM分类的时间复杂度，还可以提高SVM的分类精度，如何根据实际的训练数据选择恰当的核函数是SVM在应用中亟待解决的一个关键问题。目前常用的SVM核函数有：线性核函数、多项式核函数、Sigmoid核函数等。

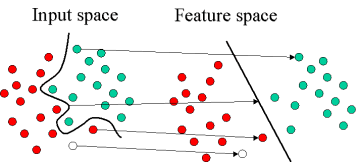


图 2‑5 支持向量机示例

支持向量机是一种有坚实理论基础的小样本学习方法。从本质上看，它高效地实现了从训练样本到预报样本的“转导推理”，不涉及大数定律、概率测度等传统的统计方法，很大程度上简化了通常的分类和回归问题。而且，支持向量的数目决定SVM的计算复杂性，而并非样本空间的维数，从而降低了“维数灾难”产生的复杂度。同时，支持向量机具有较好的健壮性，算法简单易实现。但是，SVM难以处理大规模的训练样本，在解决多分类问题时存在困难。

3.5.1.4朴素贝叶斯

贝叶斯分类是一类分类算法的总称，这类算法均以贝叶斯定理为基础，故统称为贝叶斯分类。而朴素贝叶斯分类(NBC：Naïve Bayes Classifier)是贝叶斯分类中最简单却应用最为广泛的一种方法。朴素贝叶斯分类器基于一个简单的假设：给定样本的特征之间相互条件独立。根据贝叶斯定理，给定样本特征x,该样本属于类别y的概率如式(2-1)所示：

 (2-1)

式中，x是一个特征向量，将其维度设为M。根据朴素贝叶斯样本特征之间相互独立的假设，式(2-1)可以用式(2-2)表示：

 (2-2)

通过公式2-2，求样本特征x属于类别y的概率就转化为统计类别y的先验概率问题。在给定已知的训练样本集合以及类别标签的条件下，计算出未知的样本属于每一类别的条件概率，取概率最大的一类为最终分类结果。

朴素贝叶斯分类算法逻辑简单，易于实现，且分类过程中时空开销小。理论上，朴素贝叶斯模型与其他分类算法相比有最小的误差率。但事实却并非总是如此，这是因为朴素贝叶斯模型假设特征之间相互条件独立，这个假设在实际应用中很多时候并不成立。因此在特征数目较多或特征之间相关性较大时，分类效果不好；而在属性相关性较小时，朴素贝叶斯分类的性能最为良好。

朴素贝叶斯发源于有着坚实的数学基础以及较为稳定的分类效率的古典数学理论，另外朴素贝叶斯实现还有所需的参数少、对缺失数据不敏感、实现算法简单的优点。但是朴素贝叶斯要求特征之间的相关性较小，甚至完全独立，在数据特征相关性较大时，朴素贝叶斯的分类效率较低。

3.5.1.5 K最近邻法

K最近邻（kNN，k-Nearest Neighbor）算法是一种常用的监督学习方法，其工作机制非常简单：给定测试样本，基于特定的距离度量找出训练数据集中与其最靠近的k个样本，然后基于这k个“邻居”的信息对测试样本的身份进行预测。在分类问题中一般是选择这k个样本中出现信息最多的类别标记作为预测结果，即这k个样本中属于哪个类别的样本最多，待分类样本就被判定为哪个类别。

与上述各种算法相比，KNN算法最大的不同之处在于：它没有显式的训练过程。在训练阶段仅仅是将样本保存起来，训练时间开销为零，待收到测试样本才进行处理【西瓜书】。KNN算法简单，易于理解，易于实现，适合对稀有事件分类和处理多分类问题。但是参数k的取值十分重要，当k取不同值时，分类结果会有显著差异。当k较小，分类结果易受噪声点影响；k较大时，最近邻点中又可能包含太多其他类别的点。本文的实验中，我们将k取值为30。此外，距离度量方式的选取对分类结果也有很大影响，采用不同的距离度量计算方式，找出的“近邻”可能会有显著差异，从而导致分类结果也不相同。

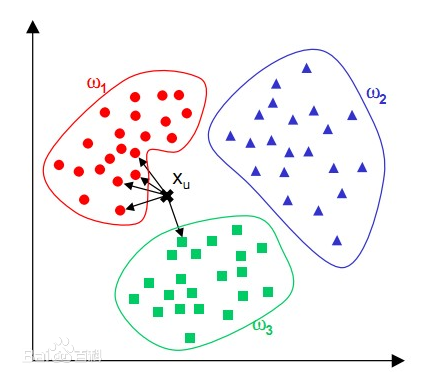


图 2‑6 k最近邻示例

3.5.2 评估指标介绍

本文使用precision和recall两个参数对实验结果进行评估。Precision（准确率，又称查准率）和recall（召回率，又称查全率）的概念源于信息检索系统，是用来衡量某一类文献检索系统的信号噪声比的指标，后来也用于评估模式识别中分类算法的性能。精确率通常用于评价结果的质量，而召回率则用于评价结果的完整性，其计算方法如下：

其中：

FN — False Negative,被判定为负样本，但事实上是正样本；  
FP — False Positive,被判定为正样本，但事实上是负样本；  
TN — True Negative,被判定为负样本，事实上也是负样本；  
TP — True Positive,被判定为正样本，事实上也是正样本。

Precision和recall值越大，说明分类效果越好。

由上述公式可以看出：准确率计算的是所有“正确被判定为正的样本”占所有“标记为正的样本”的比重；召回率反映了“正确被判定为正的样本”占所有“实际为正的样本”的比重。一般来说，准确率和召回率反映了分类器性能的两个方面，单一依靠某个指标并不能较为全面地评价一个分类器的性能。

通过公式我们也不难看出，准确率和召回率为互逆相关性。一般来说，召回率为60%~70%，准确率为40%~50%,当召回率超过70%时，若想再提高召回率就必然会降低准确率。为了兼顾精确率和召回率，定义如公式\*\*\*所示的综合评价指标F【Android设备指纹识别，P58】，我们认为精确率比召回率更重要，所以选择b=0.5，即综合评价指标为F0.5.

F值计算公式。

3.5.3 训练和测试过程

交叉验证（Cross Validation），也被称作循环估计（Rotation Estimation），是一种统计学上将数据样本切割成较小子集的实用方法。其基本思想是把数据进行分组，一部分作为训练集（train set），另一部分作为验证集（validation set or test set）。首先用训练集对分类器进行训练，再利用验证集来测试训练得到的模型（model），以此来作为评价分类器性能的指标。

十折交叉（10-fold cross validation）【沈超博士论文参考文献122】是最常用的交叉验证方法。将样本随机分割成规模相同的10个子样本，将其中9个子样本作为训练数据，一个子样本作为测试数据，交叉验证重复10次，最后取10次测试结果的平均值。这个方法的优势在于同时重复运用随机产生的子样本进行训练和验证，充分利用了所有样本。

从模式分类的角度来看，无线设备的识别是一项有挑战性的任务。在某些场景下，它是一个单分类（合法 vs 非法）问题，例如判断一个热点是否为钓鱼AP或用户登录时，仅需识别是否为合法用户的设备在进行登录；但在某些场景下，这又是一个多分类问题，例如需要具体的识别无线设备个体时。鉴于很多分类算法在原理推导上都是基于二分类的假设，在实验中需要将多分类问题转化为二分类问题，做法是把数据集中的某一类当做一类，其他的所有类当成另外一类。

我们首先选择一个设备为正例（），将其样本标记为+1，其余所有设备的数据样本标记为-1。按照如下的步骤训练并且测试分类器对无线设备的识别能力：

第一步：将正例数据和负例数据分别随机分割成规模相同的10个子样本；

第二步：将其中的9个子样本分别作为正例和负例的训练样本，建立相应的认证模型；

第三步：将剩下的1个子样本作为测试数据，测试分类器对于目标设备和其他设备的识别能力；

第四步：依次指定其他各设备为合法，并重复上述步骤得到每个设备的合法分数和非法分数。

由于在训练过程中使用了随机切割数据的方法生成训练样本，为了消除这种随机性对实验结果的影响，我们将上述实验重复了50次，每次重复独立地从样本池中选择合法的训练样本。

3.6 实验结果与分析

3.7 对比实验

3.8 本章小结

4.无线网络设备识别系统的实现与测试

前几章的研究和实验结果表明，网络流量中确有包含设备身份信息的特征指纹，可用于设备的识别与认证。基于此，我们设计并开发出基于B/S架构的无线网络设备识别原型系统，本章从需求分析、架构设计、系统实现以及功能测试四个方面对原型系统进行介绍。

4.1原型系统需求分析

本论文依托于课题组和电子信息控制重点实验室的合作项目“基于行为特征认知的无线网络目标指纹识别技术研究”，该原型系统主要用于\*\*实验室在该单位内部网络中对网络内连接的各种设备进行不间断地、被动式、实时性身份识别及认证，在设备正常使用的同时，网络管理人员可以自发性完成设备网络流量数据捕获、设备特征指纹的形成、无线设备指纹生成、设备身份识别认证等功能，该系统也可以应用于其他对安全性要求比较高的政府、企业等单位内部网络。

我们在对无线设备指纹识别技术研究的基础上，开发了一套完整的原型系统。原型系统的设计基于B/S（Browser/Client）的架构，用户从浏览器(Browser)上进入系统进行操作，数据的传输、存储、分析在中心服务器上完成。原型系统的工作逻辑如图\*\*\*所示。



系统捕获接入到局域网的无线设备网络流量，从TCP数据帧中提取帧间隔时间、帧大小和传输速率三个参数的数据，根据\*\*\*中介绍的方法提取特征指纹。在形成设备指纹后，与现有指纹库中的指纹进行比对，若该设备已在指纹库中，则识别出是哪一个设备；否则将其作为新的指纹加入到指纹库中。

原型系统主要分类网络流量捕获、特征指纹的形成、指纹库的构建、无线网络设备识别四个模块。每个模块的需求如下：

1. **流量数据捕获：**通过开发第三方python工具库，系统可以捕获接入到局域网的无线设备网络流量，并将其上传至后台服务器；捕获流量的过程对于对用户是透明的，不会干扰用户的正常使用；管理员可在web页面控制什么时候开始捕获流量，什么时候采集截止（为了区别使用系统的用户和使用无线设备的用户，我们将系统的使用者称为管理员）；管理员可随时查看数据采集的情况，如采集的设备数目和数据量等；当数据量足够时，系统提示可以结束采集，数据量过少时也应给出相应提示；管理员也可以直接上传pcap格式的流量文件。
2. 流量数据捕获：（以UML图或者流程图的形式描述用户与系统交互逻辑）流量数据捕获分为离线文件上传和在线数据捕获两种方式。用户可以通过离线文件的形式将任何一台无线设备的网络流量数据上传到系统以便对设备的身份进行识别认证；或者用户也可以在系统中实时开始采集连接入网络的设备数据，这样用户不仅可以实时观察设备的流量数据，也可以在数据规模达到一定程度时将数据保存成文件进行设备识别认证。
3. **特征指纹的形成：**后端处理捕获到的流量数据，提取帧间隔时间、帧大小和传输速率三个参数的数据，对其进行降噪和归一化处理，提取特征形成设备指纹；此功能中管理员可控制的输入有：1）降噪的阈值；2）特征指纹的构建方法，有基于概率密度的特征指纹和基于特征融合的特征指纹，如果是前者，还需确定使用三种参数（帧间隔时间、帧大小和传输速率）的哪一个；3）每组样本的大小（group\_size）和窗大小（bin\_size）；前端获取管理员的各项输入，传递给后端；后端接收参数形成特征指纹。
4. **无线网络设备识别：**将待识别的设备指纹输入到分类器中，完成无线设备模型的构建与评估，计算precision，recall和F值三项评估指标；可以设置一个评估阈值，若F值（或precision/recall）大于该阈值，则认为这是指纹库中已有设备，并识别出是哪一个；若F（或precision/recall）小于该阈值，则认为这是一个新的设备，将其加入指纹库中；管理员可以选择使用哪一个分类器（随机森林、支持向量机、人工神经网络、K最近邻和朴素贝叶斯），也可以选择使用多个分类器，系统会展示多个分类器的识别结果；分类器的部分参数也是可调的，如K最近邻的距离度量方法、随机森林的决策树数目等；前端获取管理员的输入，传给后端；后端接收参数给出识别结果再传给前端；前端接收结果并展示在页面上。
5. **指纹库的构建：**系统初始状态存储了若干设备指纹，为基准指纹库；当系统生成新的设备指纹后，与指纹库中的指纹进行比对，若该设备已在指纹库中，则识别出是哪一个设备；否则将其作为新的指纹加入到指纹库中，更新指纹库；这部分功能在后台服务器上完成。

4.2系统架构设计

根据5.1中对原型系统的需求分析，我们设计出如图\*\*所示的系统用例图和表\*\*所示的功能模块设计表。

|  |  |
| --- | --- |
| **模块名称** | **功能介绍** |
| 流量数据捕获模块 | 搭建多拓扑复杂无线网络，调用Wireshark的API编写程序实时采集实验中设备发出的流量，将生成的文件保存在系统后台；从前台可视化模块读取数据采集参数，在界面上显示数据相关信息 |
| 特征指纹形成模块 | 读取捕获的网络流量文件，从中提取各项参数；从前端页面读取阈值完成数据降噪和归一化；根据用户选择的特征构建方法形成设备指纹并保存为文件形式 |
| 无线网络设备识别模块 | 验证系统的核心模块，获取用户选择的分类器及各项参数，对设备指纹进行训练和测试，计算评估参数 |
| 指纹库构建模块 | 根据分类器的各项评估参数，确定设备指纹是否已在指纹库中；如果不在将其确定为新的指纹加入指纹库中 |

4.3 原型系统实现

原型系统采用B/S架构的模式，服务器端统一采用Web服务器的形式，数据交换基于JSON数据格式。下面分别对各部分进行阐述。

4.3.1流量数据捕获模块

4.3.2特征指纹形成模块

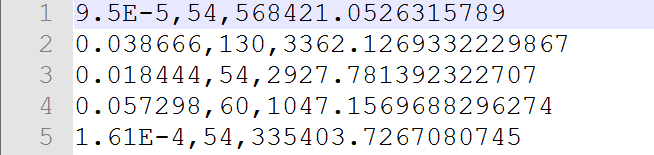
特征指纹形成模块基于\*\*节介绍的指纹提取方法开发，分为流量参数的提取、数据降噪、数据归一化、特征指纹的形成几个步骤。下面详细介绍每个步骤的前端和后端设计。（流程图）

1.流量参数的提取

后台读取pcap格式的流量文件，从中提取出每条数据帧的帧时间间隔、帧大小和传输速率三个参数，并保存为data.txt。数据格式为

X =[xiIAT，xiFS，XiTR]

其中xiIAT为第i条数据帧的帧时间间隔，为第i条数据帧的帧大小，为第i条数据帧的传输速率。数据之间以逗号隔开；具体如下图所示：



保存的结果以JSON格式（介绍下JSON格式）回传给前端，前端显示处理结果，包括散点图、柱状图和表格形式的数据（界面截图）。

2．数据降噪

后台读取data.txt文件，根据各项参数的阈值对其进行过滤，注意降噪结果只保留三个参数均小于相应阈值的样本，即便样本只有一个参数大于相应阈值，该样本也会被舍弃。结果保存为denoised.txt，该文件在用户退出系统后也会被删除。降噪的结果仍以JSON格式回传给前端，

前端接收数据并可视化，用户可查看降噪结果（界面截图）。

3.数据归一化

后端读取denoised.txt文件，采用最大值-最小值方法对每项参数的数据（即denoised.txt文件中的每一列）归一化处理，归一化结果保存为normalized.txt文件，同时以JSON格式回传给前端。

前端接收归一化数据，以图表形式展示给用户。（界面截图）

4.特征指纹的构建

前端获取用户设置的各项参数，包括：特征指纹提取方法（基于概率密度的特征指纹/基于特征融合的特征指纹）、分组大小（group\_size）和窗大小(bin\_size)，以字符串的格式传给后端。

后端接收前端传递的数据，读取normalized.txt文件，根据\*\*中所述方法提取相应特征。对于基于概率密度的特征指纹提取方法，本阶段生成三个文件：data\_IAT\_feature.txt,data\_frameSize\_feature.txt,data\_transRate\_feature.txt，分别存储每个参数生成的特征；对于融合特征指纹，将生成一个data\_mix\_feature.txt文件，存储融合的特征指纹。

（提取完特征跟前端如何反馈）

值得指出的是，为了避免随着用户使用系统次数的增加，系统数据文件越来越多占用空间的问题，上述过程中生成的data.txt、denoised.txt和normalized.txt文件均为临时文件，在用户退出系统后均会被删除。且若此设备被为指纹库中已有的设备，则四个feature.txt文件也会被删除；如果为新设备，则将四个feature.txt文件上传到指纹库中。

4.3.3无线网络设备识别模块

无线设备识别模块的功能是按照十折交叉法划分训练和测试数据，将其输入到分类器中，再计算出\*\*节中所述的几种指标。

1. 十折交叉划分训练数据和测试数据

根据用户选择的特征提取方法，读取相应的feature文件。按照每台设备的数据个数，按比例将其随机划分为十份，其中九份用于训练，一份用于测试。训练和测试数据以及其相对应的类别标签分别存储在列表train、test、train\_label、test\_label中。这一部分不涉及前端设计。

1. 五种分类器的实现及性能评估

用户在页面上选择要使用的分类器（随机森林、支持向量机、朴素贝叶斯、K最近邻和人工神经网络）；对每种分类器，部分参数是可调节的，如随机森林中决策树的个数和支持向量机中的核函数等（截图）；如果用户没有调节参数，则使用其默认值。前端将用户选择的分类器及参数传给后端。

Sklearn是python的重要机器学习库，其中封装了大量的机器学习算法。后端调用Sklearn中的随机森林、支持向量机、朴素贝叶斯、K最近邻和人工神经网络这五种算法的接口，完成无线设备指纹的训练和测试。十折交叉划分的数据中train和train\_label用于无线设备模型的构建，test和test\_label则用于测试。分类器输出predict\_label，即对于测试样本的预测标签，根据预测标签可以计算出precision、recall和F值。后端把这三项指标保存为JSON格式，传给前端。

前端显示分类器的结果（界面截图）。

4.3.4指纹库构建模块

指纹库的构建包含基准指纹库和指纹库的更新两个部分。我们先将若干设备的指纹放入指纹库中作为基准指纹库，得到新的设备指纹后再将其加入指纹库中。本文所构建的指纹库不仅为设备识别提供依据，还将作为公开的数据集，供其他的研究者使用。

1.基准指纹库

我们将本次实验中得到的13台设备指纹作为基准指纹库，这13台设备的详细信息见\*\*小节。为了标识每台设备，每个设备都有自己的ID，设备ID一般是设备型号和设备编号组成的字符串，如“iphone7\_1”。如\*\*中提到的每个设备指纹包含4个feature文件，则该设备指纹文件为iphone7\_1\_IAT.txt, iphone7\_1\_frameSize.txt, iphone7\_1\_transRate.txt, iphone7\_1\_mix\_feature.txt.

2.指纹库的更新

在上述基准指纹库的基础上，设置一个阈值。若识别结果高于该阈值，则该设备被判断为新设备。系统询问用户是否需要保存该设备指纹，若需保存则用户还需为设备命名，系统根据用户的命名再加上编号即为该设备ID。若用户在之前的步骤中已生成了设备的四个特征文件，则将文件名分别改成设备ID加上特征名，上传到指纹库，若特征文件不足，则自动补足后上传至指纹库。完成指纹库的更新。

用户可随时查看指纹库中设备指纹情况。

4.3.5web服务器端程序

服务器端的开发采用python语言，具体使用python的Django框架搭建web服务【Android设备之43】。Django采用MTV框架，其中M代表模型（model），负责业务对象和数据库的关系映射；T代表模板（Tamplate），负责把页面展示给用户；V代表视图（View），负责业务逻辑，并在适当时候调用Model和Tamplate。当web服务器收到一个http请求，Django在URL配置文件中找到对应的视图函数来处理http请求，视图函数调用相应的数据模型来存取数据、调用相应的模板向用户展示页面，处理结束后返回一个http的响应给Web服务器，Web服务器将响应发送给客户端。除了以上三层之外，还需要一个URL分发器，它的作用是将一个个的URL的页面请求分发给不同的View处理，View再调用相应的Model和Tamplate。其处理流程如图【】所示。



Django的MTV框架逻辑清晰，易于使用。下面分别从MTV的三个方面介绍具体实现。

1. 模型（model）

模型是一个抽象层，用来构建和操作web应用中的数据。模型中包含所存储数据的必要字段和行为。Django对各种数据库提供了很好的支持，本论文采用sqlite作为服务器端数据存储系统。

无线设备指纹识别系统需要4张表，分别为all\_FP,IAT\_FP,frameSize\_FP和transRate\_FP。all\_FP表对应的数据结构如表\*\*所示，记录所有设备的指纹信息，主键为设备ID，其余列表为设备的各项特征，其中2-21为IAT

1. 模板（Tamplate）

模板层提供了设计友好的语法来展示信息给用户。使用模板方法可以动态地生成HTML。Django自带的模板语言包含HTML代码和逻辑控制代码，其语法主要分为两种：{{变量}}和{% Tag %}，{{变量}}主要用于和视图变量做替换，{% tag %}主要用于做逻辑判断和实现某些功能，正因有了数据+逻辑才构成了模板语言。

模板的作用是在服务器端把变量嵌入到HTML中渲染后，返回给浏览器来达到前后端代码分离，页面动态显示的目的。

1. 视图（View）

视图用于封装负责处理用户请求及返回响应的逻辑。在视图函数中，其输入参数为HTTPRequest变量，即用户的请求。完成相应的处理任务后，这个视图会返回一个HTTPResponse对象，其中包含生成的响应。每个视图函数都必须返回一个HttpResponse对象。

视图可以看做是前端与数据库的中间人，他会将前端想要的数据从数据库中读出来给前端，也会将用户想要保存的数据写到数据库。数据以JSON格式交互，JSON格式数据处理较为简单，在服务器端直接使用python自带的JSON库即可转换格式。

5. 总结与展望

5.1 论文工作总结

在这篇论文中，我们提出了一种基于特征指纹分析的无线网络目标识别方案。在传统的设备身份认证信息容易被篡改和伪装的前提下，本文基于对无线网络中移动设备发出流量数据帧的统计分析，从无线数据帧与设备个体的相关性角度对设备的身份进行认证识别。这个思路不仅适用于无线网络移动设备的识别，也可以被用于有线网络中设备、路由器、交换机等网络个体的识别。对于网络安全防御和网络用户隐私信息的安全保护起到很大的积极作用。本文所用的方法克服了主动式识别需要被识别设备参与、回复消息或者安装第三方软件的弊端，在无需被识别设备的参与下就能完成对设备的识别工作。

本文的工作总结如下：

1）搭建了一个小型无线局域网，捕获到包括PC、智能手机、平板电脑在内\*\*台无线设备的网络流量；

2）从TCP流量数据帧中提取了三种参数的数据：时间间隔（IAT）、帧大小（frameSize）和传输速率，并对三种参数的数据分别进行降噪和归一化处理；

3）本文提出两种特征指纹提取方法：一种是计算每种参数的数据的概率密度作为特征，三个参数分别形成三类特征指纹；另一个是基于特征融合的思路生成三种独立特征组合在一起的特征向量；

4）形成特征指纹后，分别使用RF、SVM、KNN、Naive Bayes、ANN五种分类器进行设备模型的构建与评估，五种分类器的F-meature值分别为：；

5）为了进一步验证移动设备识别的有效性，我们在现有的数据集上分析了分组大小、窗大小和设备数目对实验结果的影响，实验结果表明：；

6）设计并实现基于网络流量的无线设备识别原型系统，该原型系统集成了无线设备指纹识别方案中的各个模块，且对每一步的数据处理都进行可视化处理。此外，原型系统还能够在线捕获移动设备的流量数据并对其进行识别。原型系统的实现使移动设备的身份识别从学术研究的角度向工程实用的角度逐步过渡。

5.2 不足与展望

本文提出的基于网络流量的无线设备指纹识别技术存在一些局限和不足，许多问题仍有待进一步的研究和完善。

限于实验环境和经济实力，我们实验中的数据规模较小，仅包含\*\*台设备的数据，这对于无线设备的识别是远远不够的，因此创建更大规模的指纹库来测试算法的性能是完全必要的。此外，我们对每台设备的数据采集是一次性完成的，由于设备存在器件老化问题，长时间的积累会引起器件参数的退化老化效应【射频指纹提取】，设备指纹是否能够保持较长时间内的稳定性仍是未知之数。更大规模的指纹库不仅应当包含更多设备的指纹，还应当包含一台设备在较长时间间隔内的指纹。

当前基于流量的无线设备指纹相关研究缺乏一个标准的数据库，我们考虑将采集到的原始的设备流量数据作为公开的数据集，以便更多的研究者研究无线设备指纹识别问题，同时也为我们将来提取更丰富的特征提供基础。

本文从网络流量中提取帧时间间隔、帧大小和传输速率三个参数，网络流量中也许存在其它能够反映设备身份特征的信息。我们拟在后期的研究中从流量中提取更加丰富的特征，并采用类内离散度、类间离散度和PCA主成分分析等方法对各项特征展开分析，以期达到更好的识别效果。

本文所使用的分类器均为有监督学习的分类器，即输入的样本标签均已知。